Forth2020-ESP32forth

My experience as a beginner with ESP32forth

In this document, I simply want:

- to tell you about my personal experience with ESP32forth,
- to show you what is available on the Forth2020 site to get started with ESP32forth.

My ESP32forth environment

In my case, I am using the following essentials:

- the Arduino IDE version 1.8.16 on a Windows 10 64 bit AMD PC,
- the Espressif ESP32 Board Manager version 1.0.6,
- a DOIT ESP32 DEVKIT V1,
- the ESP32forth *Version: 7.0.5*.

It is also nice to have the following to do some testing:

- a homemade LED probe to see the state of pins,
- a multi-meter to double check the LED probe tests,
- a digital memory oscilloscope (I don't have one),
- or logic analyzer (I don't have one) to see what is too fast for the human eye.

The essential of KISS for beginners with ESP32forth

ESP32forth is a very simple stand alone programming language running on ESP32 MCUs. But we must also make our efforts to make it easier as a beginner. This is the **KISS** approach: Keep It Simple Stupid. I prefer to say Keep It Super Simple because I do not like feeling stupid.

Our goal here is essentially to start to familiarize ourselves with Forth using the ESP32forth available on the Forth2020 site.

The essential of KISS for ESP32forth beginners as I experienced it:

1. minimize hardware involved,
2. minimize software involved,
3. minimize the complexity of the development environment,
4. minimize the complexity of any other support software,
5. minimize the number of things you have to learn at the same time.

We want to learn Forth, not debug all kinds of other things...

When we have learnt the basics of Forth, we could add more complexity.

The way to apply KISS

This is how to apply KISS.

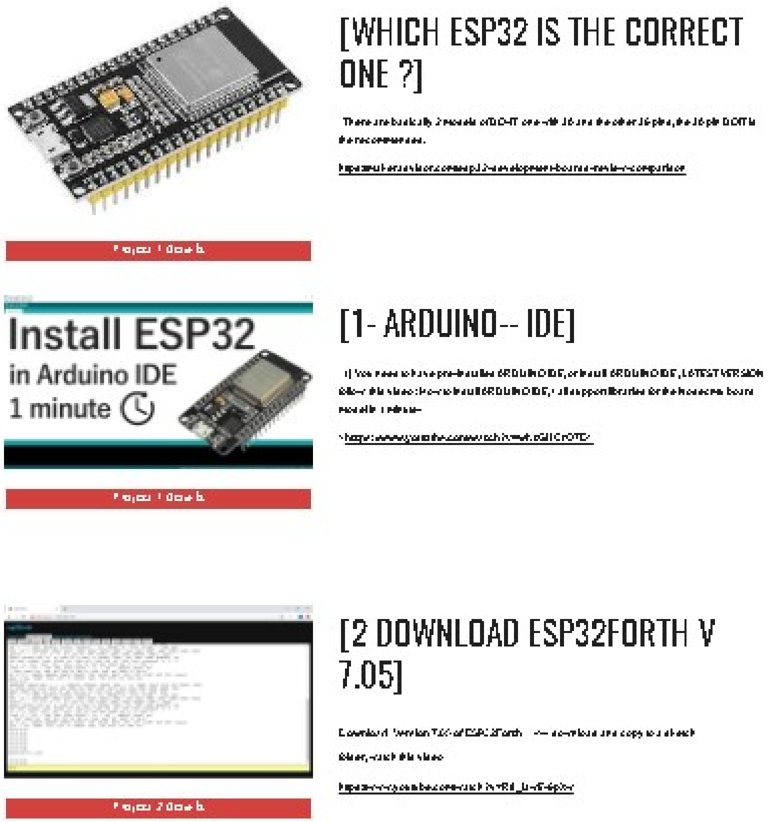
1. Use a simple Arduino IDE to install ESP32forth.
2. Use a simple serial communication port and terminal emulator to connect to the ESP32 MCU: Teraterm or Xshell.
3. Stay away initially from complex protocols like I2C, TCP/IP, and WiFi and Webservers.
4. Start with a simple software project like the **Blink** as most beginners do in any language.
5. Write the program in human sentences first because this is what humans understand best.
6. Convert the human language procedure to Forth words that express what you just said in human words.
7. Use a simple editor that you are already familiar with for all your writing.
8. Document your program with lots of comments to refresh your min later.
9. Test each word created before creating other words.

A few recommendations for yourself

Learning Forth or any other programming language takes time and effort. Take care of yourself as well to ensure you keep on having fun programming.

1. Take a pause frequently to refresh our mind which really does all the work.
2. Be patient and think positive, it takes time to learn anything well.
3. Get assistance from Forth2020 members if required.
4. Motivate yourself by sharing your experience with others.
5. Congratulate yourself when we succeed.
6. Sleep well. The inspiration might be better tomorrow.
7. **Have fun programming and learning.**

ESP32forth documentation and installation



[WHICH ESP32 IS THE CORRECT ONE ?]

There are basically 2 models of DOIT one with 30 and the other 36 pins, the 36 pin DOIT is the recommended.

https://makeradvisor.com/esp32-development-boards-review-comparison

[1- ARDUINO-- IDE]

1) You need to have pre-installed ARDUINO IDE, or install ARDUINO IDE, LATEST VERSION follow this video: How to install ARDUINO IDE, + all support libraries for the [illegible] board [illegible]

<https://www.youtube.com/watch?v=[illegible]>

[2 DOWNLOAD ESP32FORTH V 7.05]

Download Version 7.05 of ESP32Forth -> download and copy to a sketch folder, watch this video

https://www.youtube.com/watch?v=[illegible]

The Forth2020 site below provides most the ESP32forth documentation:

Forth2020-Home

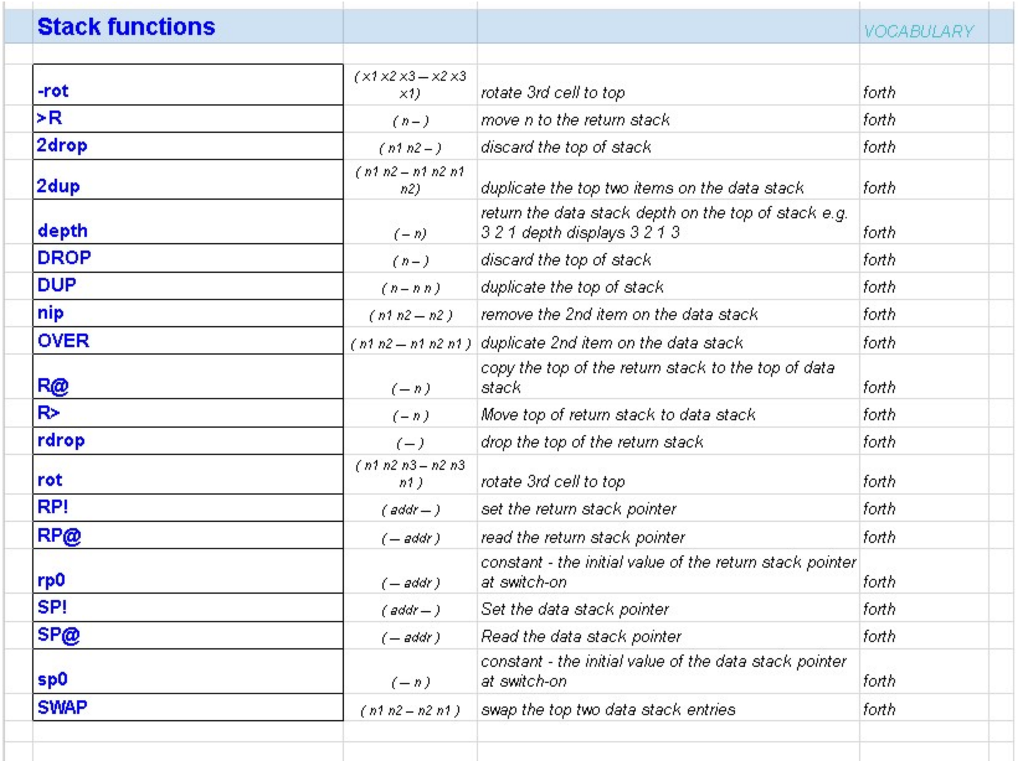
The installation process using the Arduino IDE is well documented in the **Installation** section:

Forth2020-ESP32Forth-Installation

ESP32forth vocabulary

Before we do anything with ESP32forth, we need to learn a bit of its vocabulary in the **Glossary** section:

ESP32forth-Glossary



Stack functions			VOCABULARY
-rot	(x1 x2 x3 – x2 x3 x1)	rotate 3rd cell to top	forth
>R	(n –)	move n to the return stack	forth
2drop	(n1 n2 –)	discard the top of stack	forth
2dup	(n1 n2 – n1 n2 n1 n2)	duplicate the top two items on the data stack	forth
depth	(– n)	return the data stack depth on the top of stack e.g. 3 2 1 depth displays 3 2 1 3	forth
DROP	(n –)	discard the top of stack	forth
DUP	(n – n n)	duplicate the top of stack	forth
nip	(n1 n2 – n2)	remove the 2nd item on the data stack	forth
OVER	(n1 n2 – n1 n2 n1)	duplicate 2nd item on the data stack	forth
R@	(– n)	copy the top of the return stack to the top of data stack	forth
R>	(– n)	Move top of return stack to data stack	forth
rdrop	(–)	drop the top of the return stack	forth
rot	(n1 n2 n3 – n2 n3 n1)	rotate 3rd cell to top	forth
RP!	(addr –)	set the return stack pointer	forth
RP@	(– addr)	read the return stack pointer	forth
rp0	(– addr)	constant - the initial value of the return stack pointer at switch-on	forth
SP!	(addr –)	Set the data stack pointer	forth
SP@	(– addr)	Read the data stack pointer	forth
sp0	(– n)	constant - the initial value of the data stack pointer at switch-on	forth
SWAP	(n1 n2 – n2 n1)	swap the top two data stack entries	forth

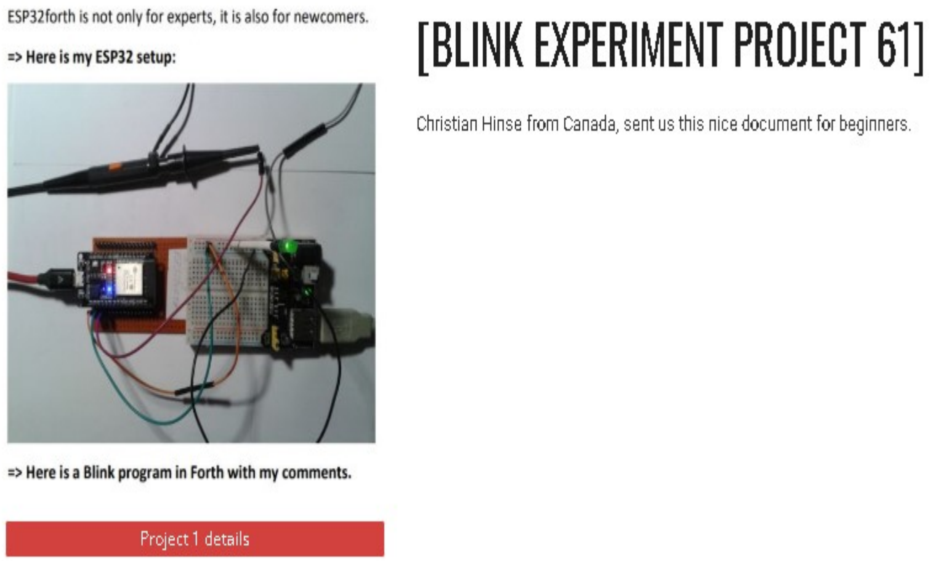
The Forth basics are:

- Word definitions,
- Comments,
- Memory,
- Stack functions,
- Maths,
- Comparison,
- Logic,
- Looping,
- Character I/O,
- Input/output.

Do not worry with the advanced hardware and software ESP32words. Just be happy that they will be available when needed.

ESP32forth experiments for beginners

ESP32forth beginners projects are shown in section **Projects 6** of the Forth2020 site. These documents are intended to go through as experiments to learn the basics of ESP32forth.



[BLINK EXPERIMENT PROJECT 61]

Christian Hinse from Canada, sent us this nice document for beginners.

ESP32forth is not only for experts, it is also for newcomers.

=> Here is my ESP32 setup:

=> Here is a Blink program in Forth with my comments.

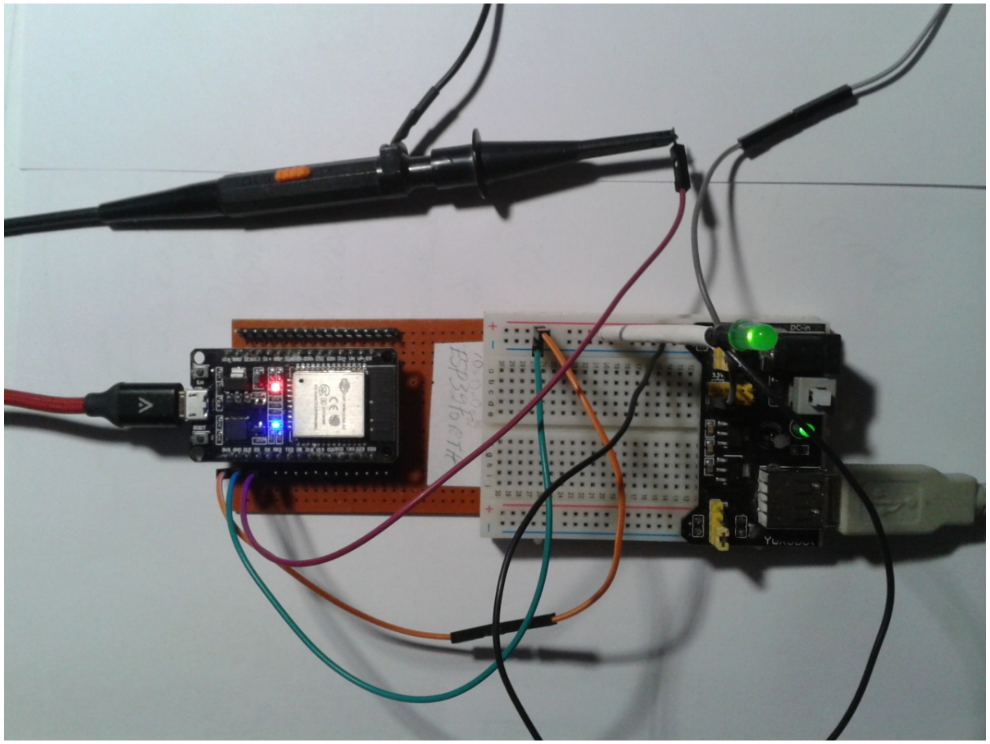
Like most beginners, I started with the simple Blink experiment.

My first ESP32forth Blink experiment

A summary of [BLINK EXPERIMENT PROJECT 61]

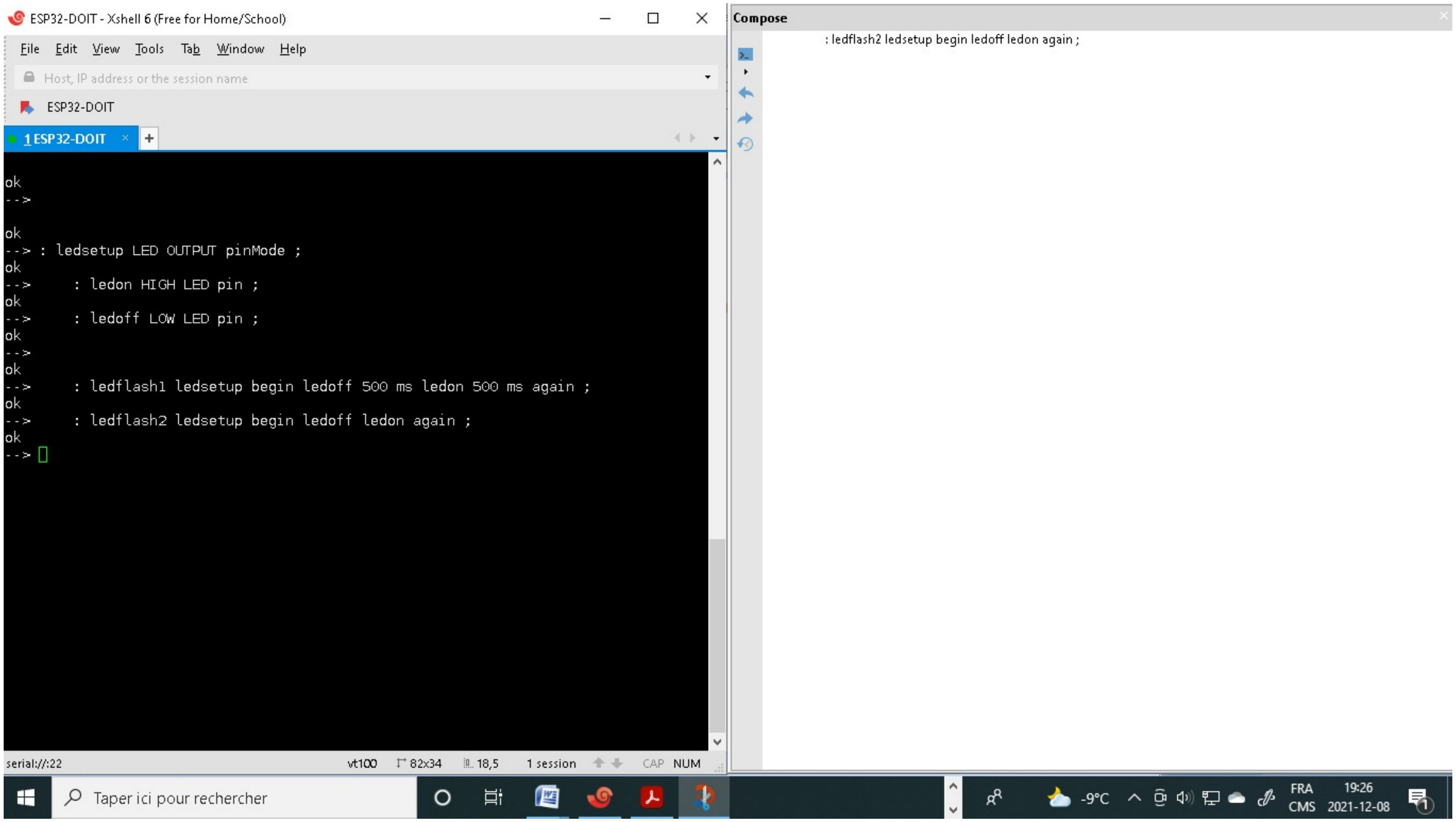
We all want to see what it looks like to use **ESP32forth**. Here is a summary of this experiment.

My DOIT ESP32 DEVKIT V1



My terminal emulator setup using the Xshell 6 terminal emulator

Here is what it looks like when I copy/paste the code text from the experiment below to my **Xshell 6 terminal emulator**. I type directly or copy/paste on the right side Compose screen from my **Notepad++** or **Office Word text editor**. The results on the left side screen.


ESP32-DOIT - Xshell 6 (Free for Home/School)
File Edit View Tools Tab Window Help
Host, IP address or the session name
ESP32-DOIT
1 ESP32-DOIT
Compose
serial://:22
vt100
82x34
18,5
1 session
CAP NUM
Taper ici pour rechercher
-9°C
FRA
CMS
19:26
2021-12-08

```
ok
-->

ok
--> : ledsetup LED OUTPUT pinMode ;
ok
-->     : ledon HIGH LED pin ;
ok
-->     : ledoff LOW LED pin ;
ok
-->
ok
-->     : ledflash1 ledsetup begin ledoff 500 ms ledon 500 ms again ;
ok
-->     : ledflash2 ledsetup begin ledoff ledon again ;
ok
-->
```

```
: ledflash2 ledsetup begin ledoff ledon again ;
```

Blink program in Forth with my comments

Preliminary information

Definitions of new Forth words start with " **:** ", followed by the new word name and a list of other Forth words defining the new word and " **; "** finally terminates the definition. In Forth, the primary location to put or get data is the **data stack**. So Forth words obtain their arguments from the data stack and leave their results on the data stack**.** Forth uses **postfix notation**: the arguments **LED** and **OUTPUT** are given before we use them with **pinMode**. You will get used to it!

1. Create a word to configure pin 2 as an OUTPUT pin.

Pin 2 is connected to an LED on the DOIT ESP32. **LED** is a predefined constant of value 2. **OUTPUT** is a predefined constant of value 1.

```
: ledsetup LED OUTPUT pinMode ;
```

2. Create words to turn the LED on or off.

The word **pin** is equivalent to the **digitalWrite** word: an alias. **HIGH** and **LOW** are predefined constants for 1 and 0. Execute **ledsetup** before using **ledon** and **ledoff**

```
: ledon HIGH LED pin ;

: ledoff LOW LED pin ;
```

3. Create a word to make the LED flash constantly.

We now use the words just created to create a new word to flash the LED. The word **begin** starts a loop. The words **500 ms** determine the duration of the ON and OFF states. The word **again** repeats the loop started at begin forever.

```
: ledflash1 ledsetup begin ledoff 500 ms ledon 500 ms again ;
```

Enter **ledflash1** on the serial communication port console to start the flashing at a frequency visible to the human eye: approximately once per second. This word is an infinite loop and you must reset the ESP32 to stop it. All previously created words are lost on reset.

A Blinking LED!

Sorry, I did not take a picture of the oscilloscope for **ledflash1.** But I have a video of the blinking LED.

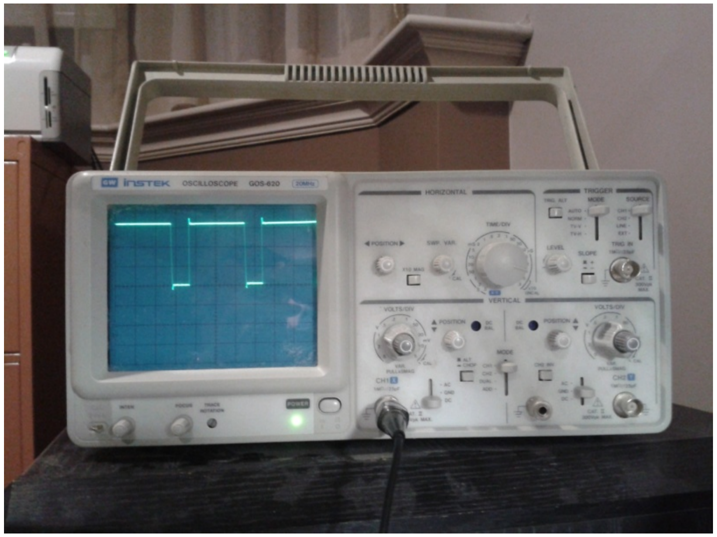
Video 1.wmv

4. Create a word to flash the LED as fast as possible.

If you want to test the maximum flash speed, you have to repeat the steps 1, 2 and 4 instead of 3.

```
: ledflash2 ledsetup begin ledoff ledon again ;
```

Enter **ledflash2** on the serial communication port console to start the flashing, but at a frequency not visible to the human eye. The included oscilloscope picture shows what it looks like. Reset the ESP32 to stop it. For the curious ones, the **frequency was approximately 575Khz**.



Now! This was simple! Yes! ESP32forth programming is easy.

Have fun with Forth...and learn the essentials of programming.

More beginners' experiments

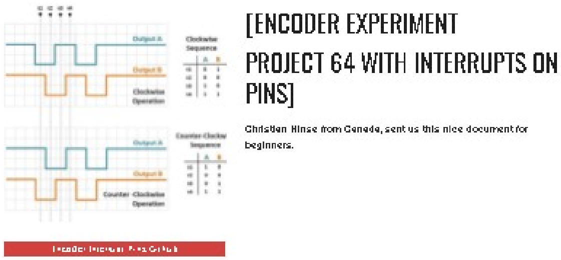
More beginner's experiments are also available from the Forth2020 site to satisfy your curiosity as you get more familiar with ESP32forth.

[TIMER EXPERIMENT PROJECT 62]

Christian Hinse from Canada told us : [illegible]

[ENCODER EXPERIMENT PROJECT 63]

Christian Hinse from Canada, sent us this nice document for beginners.



[ENCODER EXPERIMENT PROJECT 64 WITH INTERRUPTS ON PINS]

Christian Hinse from Canada, sent us this nice document for beginners.

As I learnt more about ESP32forth I became more curious and more confident.

- An experiment with timers.
- Rotary Encoder without interrupt.
- Rotary Encoder with interrupt.

More advanced ESP32forth experiments

More advanced ESP32forth experiments are also available from the Forth2020 site to challenge you a little more.

[ADDING WORDS TO ESP32FORTH PROJECT 65]

Christian Hinse from Canada, explains to us how to ADD new primitive definitions "YOUR WORDS" to ESP32forth . Thks, Christian !

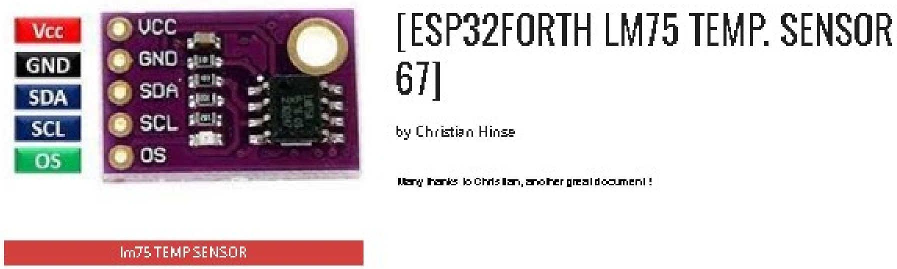
Adding Words to Esp32forth

[ESP32FORTH WIRE VOCABULARY 66]

by Christian Hinse

Many thanks to Christian, another great document !

Wire vocabulary



[ESP32FORTH LM75 TEMP. SENSOR 67]

by Christian Hinse

Many thanks to Christian, another great document !

lm75 TEMP SENSOR

I am **currently experimenting with various I2C** devices and Forth based or C based libraries for them**.** This is a challenge for me because I have to modify the ESP32forth INO file to add new words to access the C library functions.

It will help me better understand C and C++.

It is nice to be able to extend ESP32forth with Arduino libraries.

Conclusion of my experimentation with ESP32forth

My experimentation with ESP32forth brings me to the following conclusion. The ESP32forth is:

- simple to install with the Arduino IDE,
- stable at the current version,
- fast and flexible,
- motivating with its immediate interactive feedback,
- lacking a few of the features of actual languages,
- capable of extending itself easily to compensate,
- in evolution and still expanding its capacity,
- **ideal for experimentation with MCUs.**

I hope this short presentation gave you a better idea of ESP32forth.
The Forth2020 members will help you get more familiar with ESP32forth and Forth in general. I will keep on exploring ESP32forth as a beginner **and I hope to have some new beginners exploring ESP32forth with me.**
Thanks to Chuck Moore, Brad Nelson, Dr. Ting, Peter Forth, all members of the Forth2020 group and all Forth lovers.

Forth is a simple motivating interactive programming language.